Express Mail EL729266613US

METHOD, SYSTEM, AND COMPUTER-PROGRAM PRODUCT FOR PROVIDING SELECTIVE ACCESS TO CERTAIN CHILD NODES OF A DOCUMENT OBJECT MODEL (DOM)

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to the field of XML documents and, more particularly, to a method, system, and computer-program product for providing selective access to certain child nodes of a Document Object Model (DOM).

2. Discussion of the Related Art

The use of Extensible Markup Language (XML) is very popular in the development of software applications. Originally XML was used primarily for the exchanging of data between two applications, but XML is now applied to almost every aspect of a software application where easy declaration of data structures and customization of data are required.

When an XML document is read and parsed, the output of the parsing process is a tree structure called a Document Object Model, otherwise known as a DOM tree. The DOM tree is simply a tree-like structure providing a visual representation of the hierarchy of an XML document, where each node of the tree represents an XML tag.

Figure 1A is an example of a small section of XML text and Figure 1B illustrates a DOM tree corresponding to the XML text of Fig. 1A.

When parsed, the XML text of Fig. 1A will result in the DOM tree of Fig. 1B. As can be seen, the DOM tree consists of several nodes: "Customer"; "Last-Name"; "First-Name"; "Address"; "Street"; "City"; "State"; and "Zipcode", and each node may have "children". For example, the node "Customer" is a parent node to three children: "Last-Name", First-Name", and "Address". Similarly, the node "Address" is a parent node to four children: "Street", "City", "State", and "Zipcode".

Each node typically is an instance of an object and there are executable methods that may be performed on each node. For example, if the method "getChildNodes()" is called on the "Customer" node of Fig. 1A, a list is compiled containing the nodes (Last-Name, First-Name, and Address). A node may also have a text value, which may be retrieved using the method "getNodeValue(). Thus, if the "Street" node contains the value "8008 Greely Court", then calling "getNodeValue() on the "Street" node will retrieve (8008 Greely Court).

The World Wide Web Consortium has developed a standard framework and API that defines access to the nodes of a DOM Tree (see http://www.w3c.org for further information and a complete discussion of the API and framework). The w3c DOM API allows the children to be searched, enumerated (to enable retrieval of contents, the formulation of lists of children, and the like), and in some high performance DOM

implementations a child node may be looked up by the value of an attribute (which allows a program to access the contents of a DOM in any contextual manner).

Under the DOM standard of the prior art, there is no way to restrict access to certain of the child nodes while allowing access to others; it is an all-or-nothing proposition. Thus, under the DOM standard of the prior art, restricted access selectively among nodes (to allow, for example, limited access to nodes that have sensitive information or nodes that have contextual data which could only be accessed if the system is set to operate in that context) cannot be facilitated. Accordingly, it would be desirable to have a DOM in which the DOM nodes are enhanced to provide XOR access, that is, where the DOM nodes can be identified by a particular value and where only those nodes that match a selected one of these values will be "visible" at any point in time.

SUMMARY OF THE INVENTION

The present invention provides a unique method, system, and computer-program product for providing selective access to (and selective exclusion from) certain nodes of a DOM tree. Each DOM node is provided with a naming mechanism; in a preferred embodiment each node is assigned a permanent name using an XML attribute NAME to identify each node, and each node is also assigned a "context state value" using an XML attribute "CONTEXT". Changes to the operating context of the DOM tree affect the accessibility of each node in the tree. By correlating the "CONTEXT" attribute of

a child node (or nodes) to the "CONTEXT" attribute of a parent node, the correlated child node is accessible by the parent.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1A illustrates an example of a DOM tree and Figure 1B is an example of a small section of XML text corresponding to the DOM tree of Fig. 1A;

Figure 2 is a DOM tree configured in accordance with the present invention;

Figure 3 is a DOM tree illustrating the DOM tree of Figure 2, but showing the effect of correlating CONTEXT nodes in accordance with the present invention;

Figure 4 illustrates a preferred embodiment which introduces a CONTEXT attribute value entitled the "INHERIT" value;

Figure 5A is a textual representation of the XML tagging for a first example illustrating the present invention, and Figure 5B is a DOM tree representing the XML tagging of Figure 5A, and Figure 5C illustrates the "effective" XML tagging of the DOM tree of Figure 5B;

Figure 6A is a DOM tree illustrating an alternate configuration of the DOM tree of Figs. 5A-5C, and Figure 6B illustrates XML tagging corresponding to the DOM tree of Figure 6A;

Figure 7A illustrates XML tagging for an example showing the "encryption method" in accordance with the present invention, and Figure 7B is a DOM tree corresponding to the XML tagging of Fig. 7A;

Figure 8A is an example of the effective XML text of an alternate configuration of the DOM tree of Figure 7A-7B, and Figure 8B is a DOM tree corresponding to the XML text of Fig. 8A; and

Figure 9A is a textual representation of the XML tagging for an alternative embodiment of the present invention, and Figure 9B is a DOM tree described by the XML text of Fig. 9A.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In the drawings, the same reference numerals are used to indicate the same elements.

Figure 2 is a DOM tree configured in accordance with the present invention. In Figure 2, the CONTEXT value for each node is identical, i.e., "Customer". When the CONTEXT value for each node is identical, the DOM tree operates identically to the prior art DOM tree illustrated in Figure 1, that is, all nodes in the tree are accessible.

Figure 3 is a DOM tree identical to the DOM tree of Figure 2, except that, in Figure 3, the "Last-Name" node and the "Address" node have their CONTEXT values set to "Other" instead of "Customer". In accordance with the present invention, only the children nodes of a parent node (e.g., the "Customer" node) having a CONTEXT value the same as the parent node are visible (i.e., accessible); the other nodes (in this example, the "Last-Name" and "Address" nodes) are hidden (i.e., inaccessible). Further, in accordance with the present invention, the "grandchild" nodes ("Street",

"City", "State", and "Zipcode") are also hidden, since they are hidden by their parent (the "Address" node); their CONTEXT value is immaterial in such a case.

In each of the above examples, the child nodes are accessible when their CONTEXT value or attribute is identical to that of their parent; this relationship is referred to herein as a "correlation" between the CONTEXT of the parent and the CONTEXT of the children, and a child node so correlated is referred to herein as a "correlated node."

While in the above example the correlated nodes are assigned the identical CONTEXT value as that of their parent, it is not necessary that the CONTEXT values be identical for the correlation to exist. Figure 4 illustrates an alternative embodiment which introduces a new CONTEXT attribute value entitled the "INHERIT" value. The INHERIT value of a child DOM node instructs the child DOM node to adopt the CONTEXT attribute value of its parent DOM node. Thus, the parent's value for the CONTEXT attribute will determine which of its children and grandchildren, etc. to hide or allow access to. Referring to Figure 4, it can be seen that "City", "State", and Zipcode" have the CONTEXT tag "Other"; the "Street" node has the CONTEXT tag "Detail"; the "Last-Name" and "First-Name" nodes have the CONTEXT tag "Name"; the "Address" node has the CONTEXT tag "INHERIT"; and the parent node "Customer" has the CONTEXT tag "Other".

Using the above-described operations, the "Last-Name" and First-Name" nodes will be hidden, since neither has a CONTEXT tag matching that of their parent node

"Customer". However, since the "Address" node has the CONTEXT tag "INHERIT", it takes on (inherits) the CONTEXT tag of its parent, and thus takes on the CONTEXT tag "Other". Thus, the "Address" node is a correlated node, correlated to parent node "Customer." This leads to the hiding of the "Street" node (since its CONTEXT tag ≠ "Other") and to the availability of the "City", "State", and "Zipcode" nodes (since their CONTEXT tags = "Other"). In other words, the "City", "State", and "Zipcode" nodes are correlated to the "Address" node and also to the "Customer" node. Within the structure of a DOM tree utilizing the teachings of the present invention, several nodes within a descendant path in the DOM tree may take on the CONTEXT attribute value of "INHERIT" thereby enabling all nodes in the path to have their CONTEXT attribute controlled by the CONTEXT attribute value of the highest ascendent node in the path (the parent node "Customer" in this example).

The present invention teaches novel methods for restricting access to nodes based on context. The actual methods for restricting (or allowing) access to the nodes can be any of several methods; the novelty lies in evaluating the context of the nodes and using this evaluation as the basis for node restriction. For example, a node could simply be made unavailable when it does not possess the context of its parent, e.g., by saving a reference to the node in a private area of the parent node. Alternatively, all data for nodes to be hidden could be encrypted; thus, while the data itself would be "accessible" it could not be understood and thus could not be used for any purpose. Further, instead of encrypting data when it is determined to be inaccessible, all data

could be encrypted whenever a child is added to the tree, and when the CONTEXT tag was to allow access to a node, only then would the data be decrypted. The methods for making a node inaccessible to a parent node, and for encrypting data for a node, are well-known in the art; however, selective inaccessibility or encryption of nodes based on a CONTEXT value or other similar criteria is novel.

The following examples illustrate various aspects of the present invention. For the purpose of these examples, assume that there is a user's manual written in XML, and one half of the manual is a section named "Novice" which contains help information on pages 1 and 2 for a novice user, and a second section is called "Expert" which contains help information on pages 3 and 4 aimed at a more experienced (i.e. expert) user. Note that the portions of the DOM tree illustrated in broken lines indicates hidden (i.e., inaccessible and/or encrypted) nodes.

Example 1

The first example illustrates the basic operation of the CONTEXT tags. The textual representation of the XML tagging for this example is shown in Fig. 5A. Figure 5B is a DOM tree representing the XML tagging of Figure 5A. In the DOM tree of Figure 5B, the CONTEXT tag for the parent node (UserGuide) is set to "Novice". Accordingly, the "effective" DOM tree would be only the Novice branch of the tree, corresponding to the "Easy Help" chapters, and the effective XML tagging would be as shown in Fig. 5C.

Alternatively, if it was desired to allow access to only the "Advanced Help" chapters, the CONTEXT tag for the parent node could be set to "Expert" as shown in the DOM tree of Figure 6A; this would cause the "effective" DOM to be only the "Expert" branch of the tree, corresponding to the "Advanced Help" chapters, as shown by the XML tagging illustrated in Fig. 6B.

Example 2

In this example, the encryption method is illustrated. Here, the name of each Chapter is encrypted when the node is added, so that the text version of the DOM tree appears as shown in Fig. 7A, with a corresponding DOM tree as shown in Fig. 7B. Note that since the CONTEXT value is left blank and the nodes of the branches are not, the entire DOM tree is inaccessible. If the <BOOK> tag is set for NOVICE, the effective XML text of the DOM tree for this text would be as shown in Fig. 8A, with a corresponding DOM tree as shown in Fig. 8B.

In this example, when a child node with a given name and context value is added to the DOM tree, the values of any data fields in the node are encrypted. The encryption key is selected to be a private formula based on context and name, but not merely on the context and name values, so that attempts to decrypt with these values directly in any way will fail. Access to all children nodes is allowed at all times, but only the nodes that match the current context value of the parent node will be decrypted, rendering the data values in those nodes accessible and useable. When the

context value of a node is changed, then the children list that matches the context is decrypted. Children nodes that are encrypted also hide their children to maintain the security of child descendant data. So by using the encryption method, all nodes are "accessible", only those nodes with decrypted data will be useable.

Example 3

In this example, two nodes with the same name are used. While under the previously described examples, having two nodes with the same name would be considered an illegal action, by setting the context of the parent node to the same context as one of the child nodes, and by requiring that none of the child nodes share an identical context, only the node having the same context as the parent node will be available. In other words, the value of the context attribute of the parent determines which single child node is available. This can be beneficial to cover instances where, for example, the same name is used for different nodes to facilitate searching or filtering, but where it is still desirable to be able to selectively access one node branch to the exclusion of all others.

Figure 9A is a textual representation of the XML tagging for the above-described example, and Figure 9B is a DOM tree described by the XML text of Fig. 9A. In this example, since the context of the parent node, named "User Guide" is "Easy Help", only the left side, i.e., the node named "First Chapter" also having the context "Easy Help" is available, thereby giving access only to nodes "Page 1" and

"Page 2". By changing the context of the node named "User Guide" to "Advanced Help", access to the left node will be denied and access to the right node will now be allowed, thereby giving access to advanced help page 3 and page 4. This example thus illustrates how sibling nodes of the same name can be hidden or shown based on their context value.

The invention being thus described, it will be obvious that the same may be varied in many ways. Such variations are not to be regarded as a departure from the spirit and scope of the invention, and all such modifications as would be obvious to one skilled in the art are intended to be included within the scope of the following claims.